Reciplease Milestone 3

Clients: Jimmy Theis and Bryan Watts

This is the third of a series of requirments and specifications documents for the Reciplease system.

Alex M. Petitjean, David A. Savrda, Rober J. Wagner, and Robin A. Cooper. 10-16-2012.

Contents

Executive Summary

We are developing a clean, simple, and straightforward web service for storing, sharing, and following recipes. This service will offer a simple and stylistic alternative for the current available systems. By client request we are using Ruby on Rails deployed on Heroku. The output of this project will be a single website reciplease.net, that has two main abilities that set it apart from other similar systems. First and foremost reciplease.net will be a completely modern website with nice styles, simple interfaces, and cutting-edge browser features. Secondly Reciplease will also be a web service with an API for developers, because we want we want recipes to be part of everything we do. By doing this we want our recipes to be sharable on social networking sites such as Facebook. Additionally Reciplease will be fully supported on mobile devices so that we can access recipes anywhere.

These documents contains background information on the project such as client backgrounds, user needs, current solutions, and a list of features. It also contains other useful information such as use cases, storyboards, non-functional requirements, feature mapping, constraints, prototypes, change control plans, testing plans, Ruby coding standards, usability studies, and interaction architecture.

Introduction

This documentation contains all of the documents for the development of the Reciplease Web Application. In part these documents contains the features of the product the features of the product, the use cases developed and the test cases based upon them. The usability test and their results conducted in the attempt to determine the usability of the system where also included. Initial back end designs such as data flow diagrams were also included within these documents. These doucments have been used to develop a prototype used in the usability testing, the results of which have also been included within.

Background

Need

Discovering recipes

When people want to cook things, they need to learn how to cook them. Recipes are the generally accepted solution to this problem. They are exchanged on paper among friends and family, but sometimes friends and family don't cook and that makes new recipes hard to find. The internet is a much more robust way to share share information. Reciplease will allow any user to discover a wide variety of recipes in an quick and efficient manner.

Personalizing existing recipes

Recipes change as they are passed around depending on the personal preference of the cook. When someone saves a recipe for later use, they often take notes about how the resulting food turned out so they can make improvements. For paper and pencil cooks, this involves literally writing a note. Even if a recipe is found online, this usually involves printing it out or writing notes.

Collecting recipes

When people like a recipe, they usually save it later. This usually involves printing it out or writing down, then filing it away. This same activity can be done digitally with greater ease and organization.

Sharing recipes

When people like recipes, they often share them with their friends and family. Digital sharing is efficient because making copies is effortless.

Features

ID	Feature	Status	Priority	Risk	Stability	Target Release	Assigned	Reason
F1	View public recipes	Approved	High	Med	Med	0.1	Dev Team	The core idea of the site, need users to be able to publish recipes
F2	Private and unpublished recipes	Approved	Med	Low	Med	0.4	Dev Team	We don't want to clutter the site with unfinished recipes, so this allows them to hide unfinished ones
F3	Forking recipes	Approved	High	High	Low	0.6	Dev Team	The unique feature of Reciplease allows users to create their own version of a classic
F4	Following recipes and users	Approved	High	Low	Med	0.5	Dev Team	Allows someone follow a type of food or particular user they like
F5	Comments and Ratings	Approved	Med	Low	High	0.8	Dev Team	Users can give feedback
F6	Picture Linking	Approved	Med	Low	Med	0.7	Dev Team	A good visual element to the site to give visual representations of recipes
F7	User Profiles	Approved	Med	Med	Low	0.3	Dev Team	Allow users to show off their collective body of recipes
F8	Reciplease API	Approved	High	Med	Med	0.0	Dev Team	This will allows for open development
F9	Social Networks	Approved	Low	Low	High	1.x	Dev Team	Many users enjoy social networks
F10	Recipe Lookup	Approved	High	Low	High	0.2	Dev Team	Users need to be able to search for recipes

Clients

Jimmy Theis and Bryan Watts are both recent graduates of Rose-Hulman. They are both currently working at Interactive Intelligence in Indianapolis. They are dissatisfied with the current tools for online recipe builders. Jimmy's background in web development and Brain's background in API development give them a strong foundation for how a site like this can be built to satisfy this demand.

Current System

Jimmy and Bryan started this project a couple of times, but never had time to finish it. They would like to start from a clean slate and use newer technologies.

User Profiles

Amateur cooks

Amateur cooks are users who don't have much cooking experience, but are interested in learning or augmenting their cooking ability through the site as a hobby. Their technical sophistication will probably range from medium to high. They stand the most to gain from automated tools. Amateurs will enjoy sharing their accomplishments through social networking.

Experienced cooks

Experienced cooks are knowledgeable about cooking. They probably have existing methods to manage and find recipes. Their technical sophistication can be at any level. Since they are already familiar with cooking, automated recommendations will probably annoy them more than they will help them.

Admins/Moderators

Admins and Moderators will want to be able to view the site's status and delete spam and malicious users. Admins also enjoy when their site is populated with content, so regular users should be encouraged to add it and have an easy time doing so.

Developers

Developers are technically savvy users that want to use data on the site for their own applications. They need an API to effectively do this.

Stakeholder Profiles

Stakeholder	Role	Success	Failure
Jimmy Theis & Bryan Watts	Client	The project becomes another source of income via ads or paid accounts.	Time is lost managing the project.
Team Reciplease	Developers	The team gains a potential employment opportunity.	Lower grade for junior project.

Functionality not in Use Cases

1. Log In
    - The user must be able to log onto the Reciplease website
    - Users will input there account name and password into the account name and password boxes respectively to log in
2. Save Recipe
    - The user will be able to save recipe locally on their machines
    - User will be able to click the save recipe button

- User will select a file location and hit ok to save file at location

Usability Requirements

The software must be easy to use. Cooks who are unfamiliar with technology should be able to use the site with a minial learning curve. The primary functions of the site should be discoverable from the home page. Visual clutter should be kept to a minimum. All functionality except navigation should be related to the user's current view. The site should be at least as easy to use as YouTube, another site based around creating and viewing user generating content. User tasks should be simplified to as few steps as possible, with reasonable defaults. Registration should take between 1 to 3 minutes. A search should take less than 10 seconds. Creating and editing a recipe should take between 1 and 5 minutes, depending on the complexity of the recipe. Forking a recipe should take 1 to 5 minutes after locating the relevant recipe depending on the complexity of the change.

Performance Requirements

As a model for Reciplease, we used data from Recipelabs.com, which we believe is a comparable site. It is small, new and has some similar design aspects to it. For the response time, I used webwait.com to benchmark page loads on Recipelabs.

Response Time

Average transaction time (loading a page): .5 seconds
Maximum transaction time: .9 seconds

Throughput

Assuming roughly 10,000 pageviews a day, we should expect 6.94 transactions per minute, or .115 transactions per second on average. Our system should easily be able to handle more than one transaction per second.

Capacity

The server should be able to accommodate 2,000 unique users per day.

Degradation Modes

During an equipment failure or malfunction, the system should still support searching and lookup, but other functions (recipe creation, forking) can be turned off.

Reliability Requirements

Availability

The key functions associated with viewing users and recipes should reach 99 percent up-time. We expect the site to be able to handle searches and recipe user lookups without any trouble, but we may see some downtime for scheduled maintenance. We will determine the best day/time to run the maintenance based on when the user base is least active. We do not expect the more complex features like forking to have this same high level of availability, for the duration of this project we would expect to see availability of these tools sit around 95%.

Mean time between failures

The team anticipates that the mean time between crashes in the system will be two weeks (This timeframe will increase over time, and may be shorter in the earlier stages of the project).

Mean time to repair

In a situation where the system crashes and the basic functionality is no longer supported, the team anticipates a mean downtime of 1 hour. Its hard to anticipate this number, but resolving the issues to the point getting basic recipe viewing back online should not be time consuminhg. For the more complex components the MTTR is anticipated to be 3 hours.

Accuracy

This is not a major issue for the Reciplease project, as there is only on function for which accuracy will have an effect. The searching tools need to be well built so that the user is returned good matches to their input. This tool is the only one for which accuracy holds an effect.

Maximum bugs, or defect rate

The team will try to match the basic industry average of 8 defects per KLOC.

Bugs per type

Of the average 8 defects per KLOC we expect roughly 6.5 of these to be minor bugs (display/formatting issues), about 1 will be significant (bad searches, failure to load data on a page), and <.5 will be critical (data loss on submission, recipes being inappropriately deleted).

Supportability Requirements

Expansion and social integration

In order to allow developers to easily build on / make calls to the Reciplease system, the team must build an API. The API will allow easy expansion onto social networking sites, and will make tasks such as sharing recipes across the web easier. Although this adds a large degree of complexity to the early portions of the project, it will make adding smaller features on at the end very easy.

Average response time

Our team anticipates the average response time for adding a small feature on the system to be roughly one day. The team expects that adding on a more complex feature, such as a new social networking app, will take 4 to 5 days. A very complex enhancement, such as concurrent recipe editing or real-time recipe instructions, will take 2 to 3 weeks.

Coding standards

To improve modifiability, the team will follow common coding conventions for Ruby. We will define our standards based on this, and create a consistent environment that any developer could easily navigate through and add to our code base.

Hardware and Software Interfaces

Hardware

The scope of this project is completely software

Software

The system requires a database interface that supports quickly searching and retrieving potentially large amount of data. As a web app, the server must communicate using HTML. The client has requested an API that covers basic create, read, update, delete operations as available on the website.

Documentation Requirements

The documentation of this project should include

- The purpose of the website
- The people who will be using the site
- Use Cases
- User Interface Design
- Testing Process and Plan

Installation Requirements

As a web service this project has no requirements for installation, though it will require users to have a web browser to view the site.

Legal Requirements

1. Adhere to Priority 1 of the Web Accessibility Guidelines set out at W3C
2. Display a Privacy Policy informing the user what the site does with user data, and the site conforms to the The Data Protection Act. The Privacy Policy needs to explain what cookies the website will create and what they are for
3. Require user consent to leave cookies on the users machine, unless the cookie is a necessary requirement for the website to function

Design Constraints

There are several important design constraints for this project. The first is that the project must be developed in Ruby and utilize Rails. The second is that the product be hosted on Heroku. This means what ever we develop must be able to properly function on the hosting services servers. The website must be supported by Chrome, Firefox, Safari, and IE 8 and above.

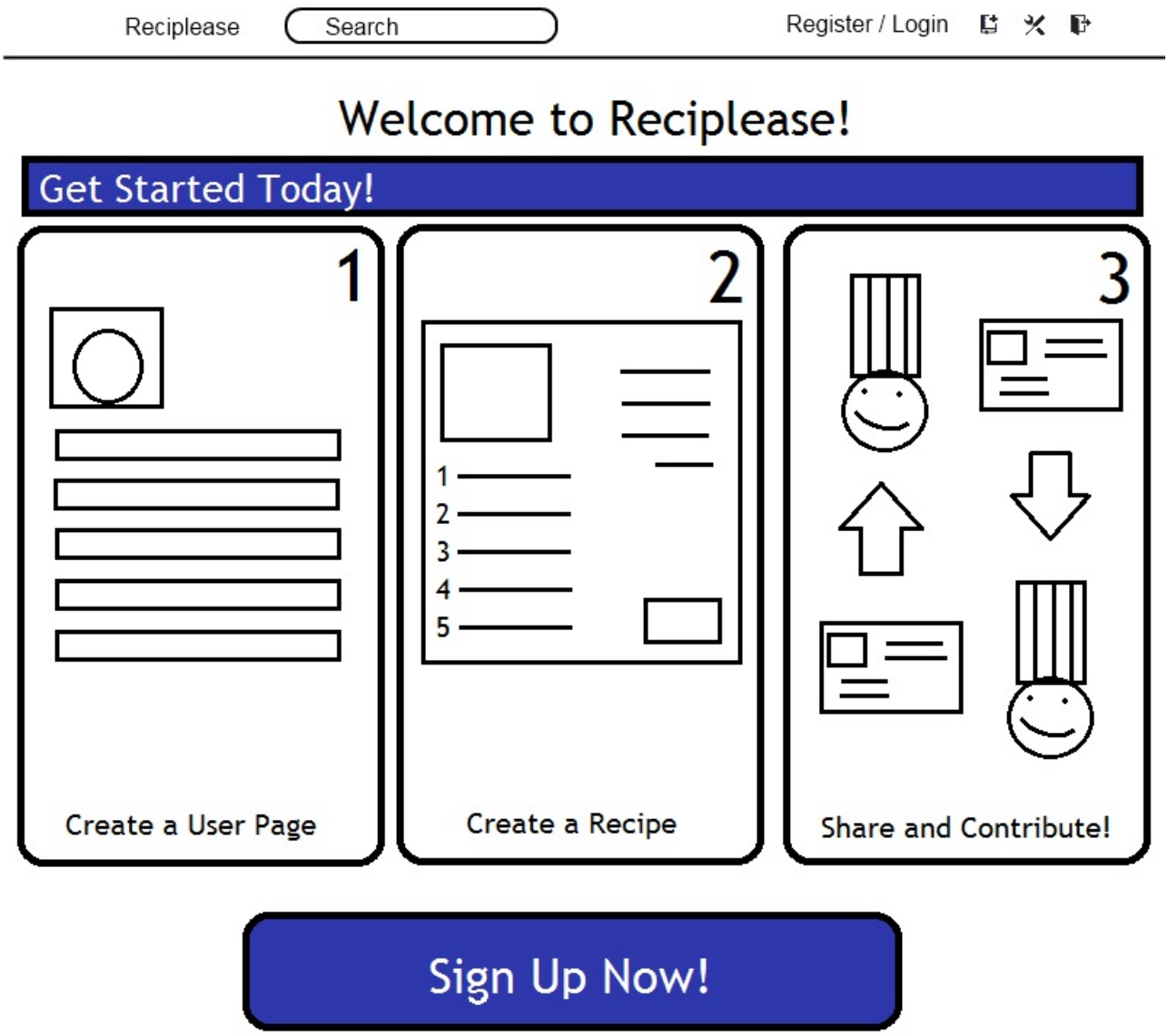
Licensing Requirements

The site must have a valid fully qualified domain in the form of xyz.domain.ext

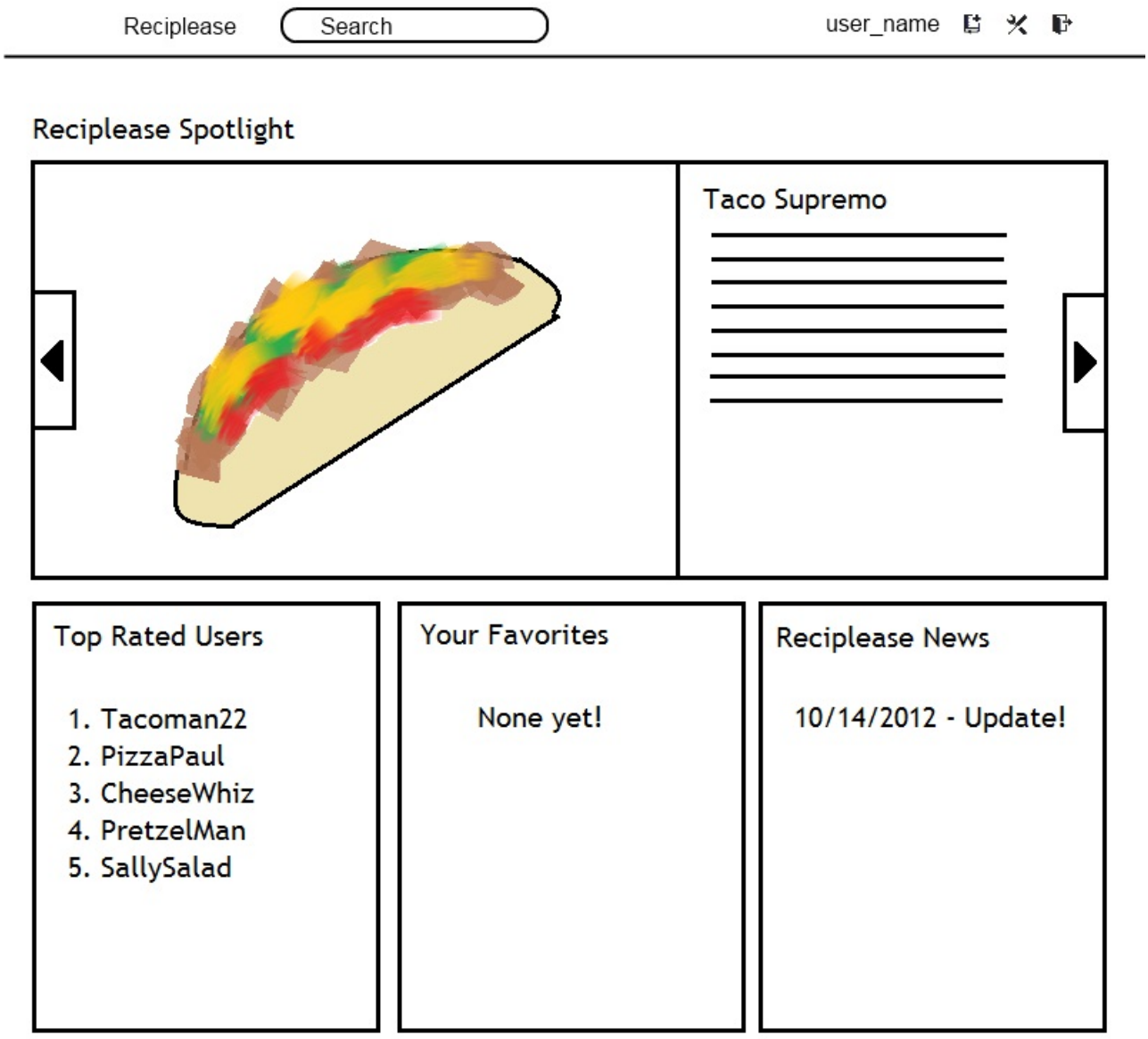
User Interfaces

Home Page

Home Page Logged Out

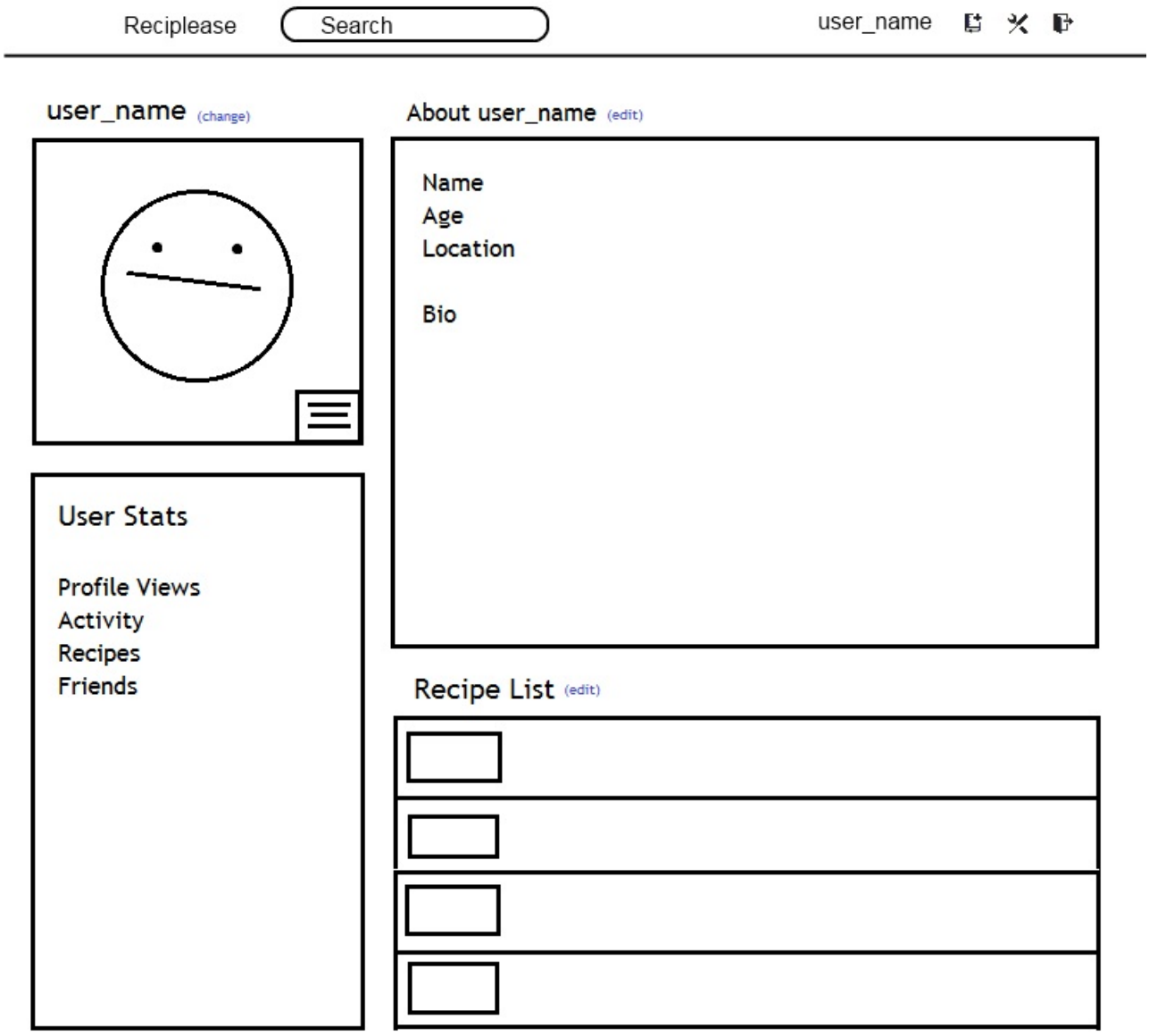


Home Page Logged In

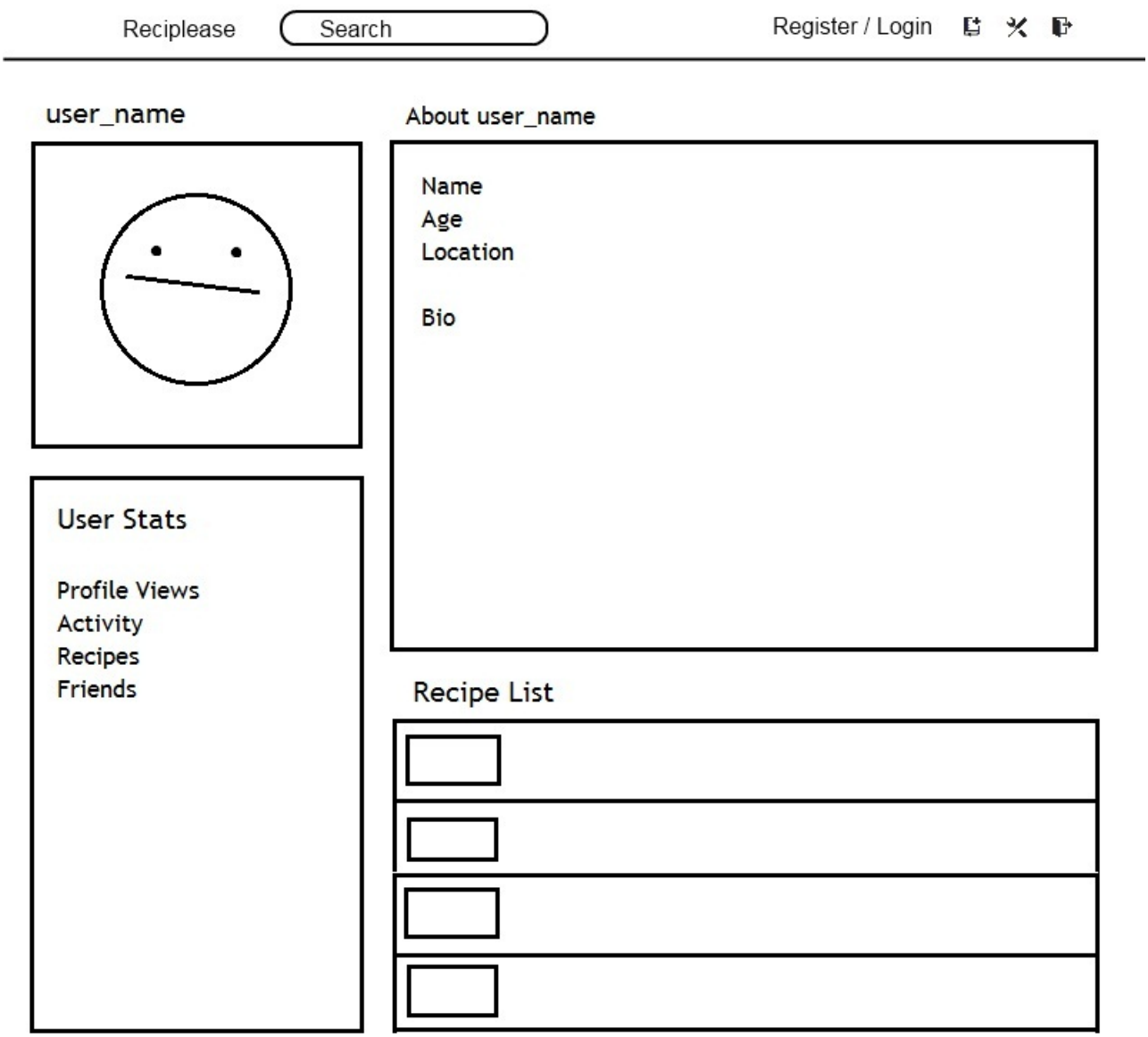


User Profile

Personal User Profile

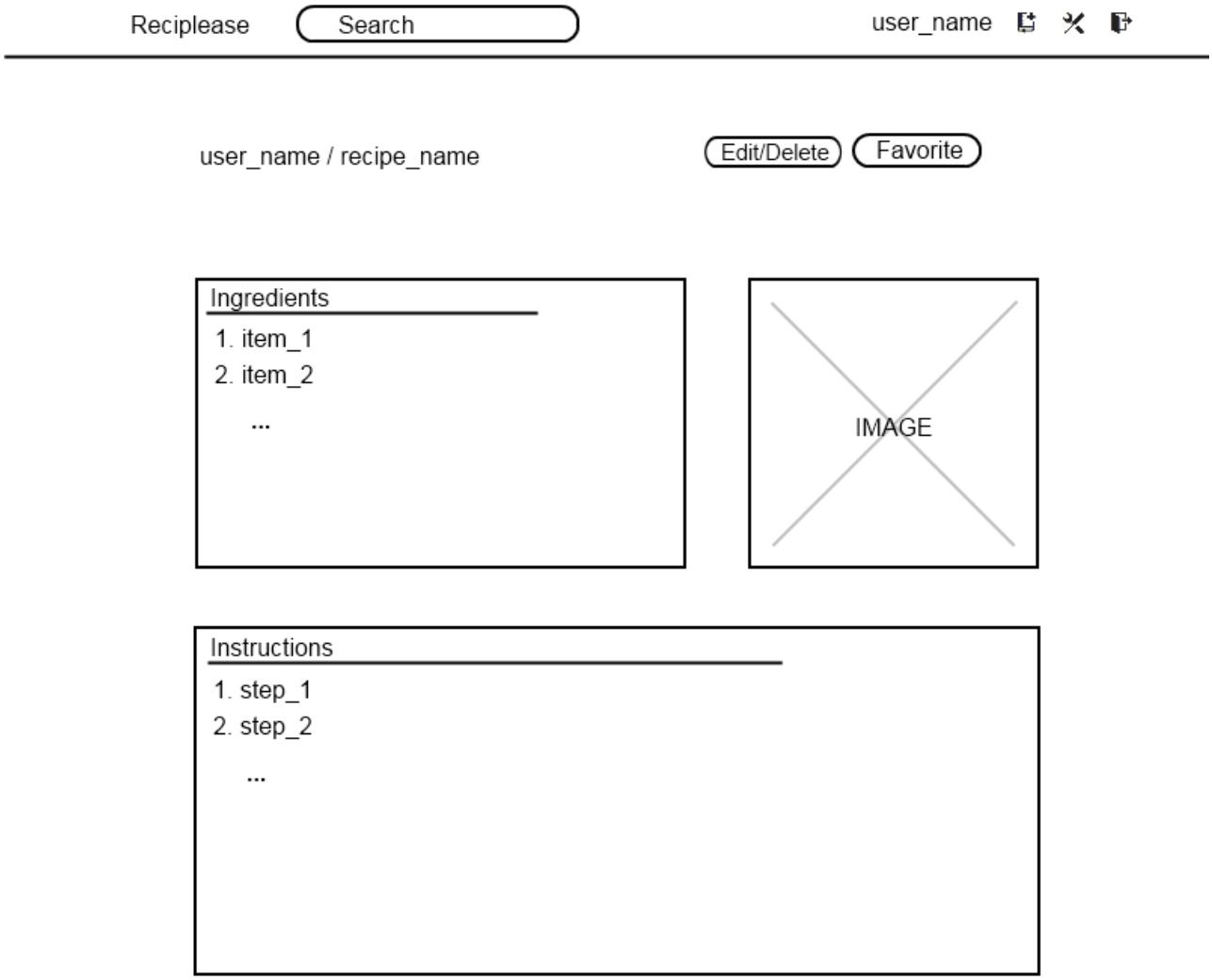


Other User Profile

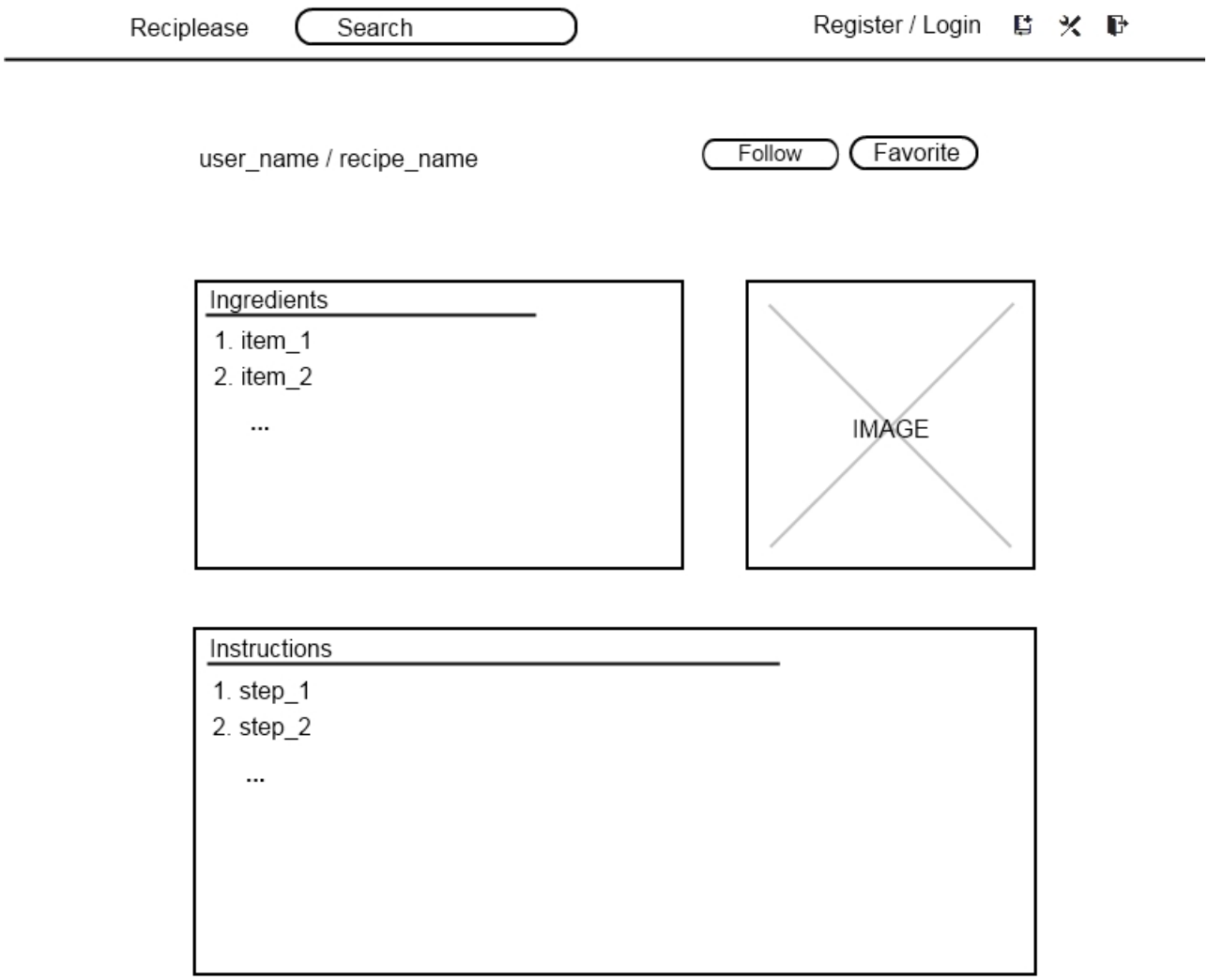


Recipe

Owner Recipe

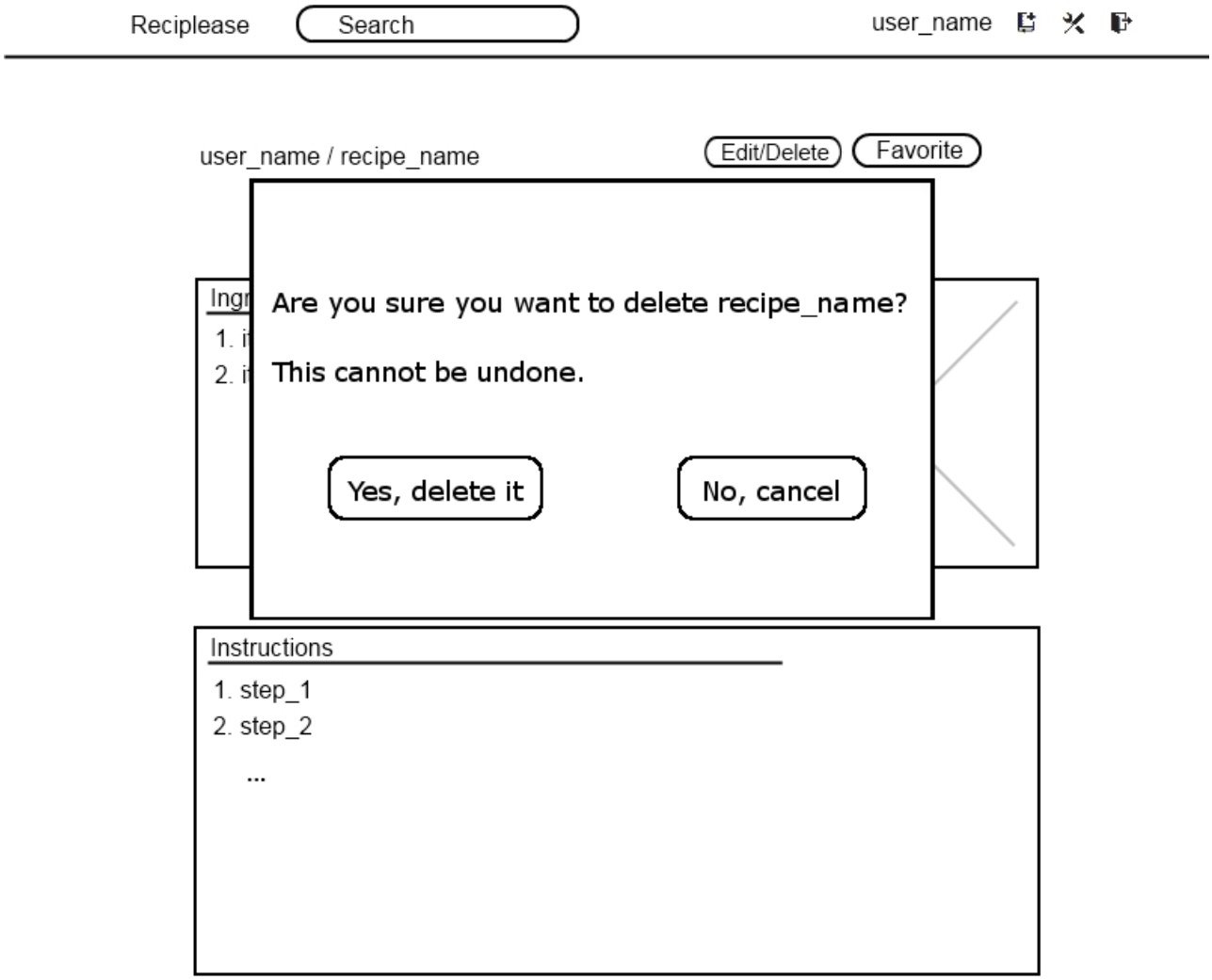


Other Recipe

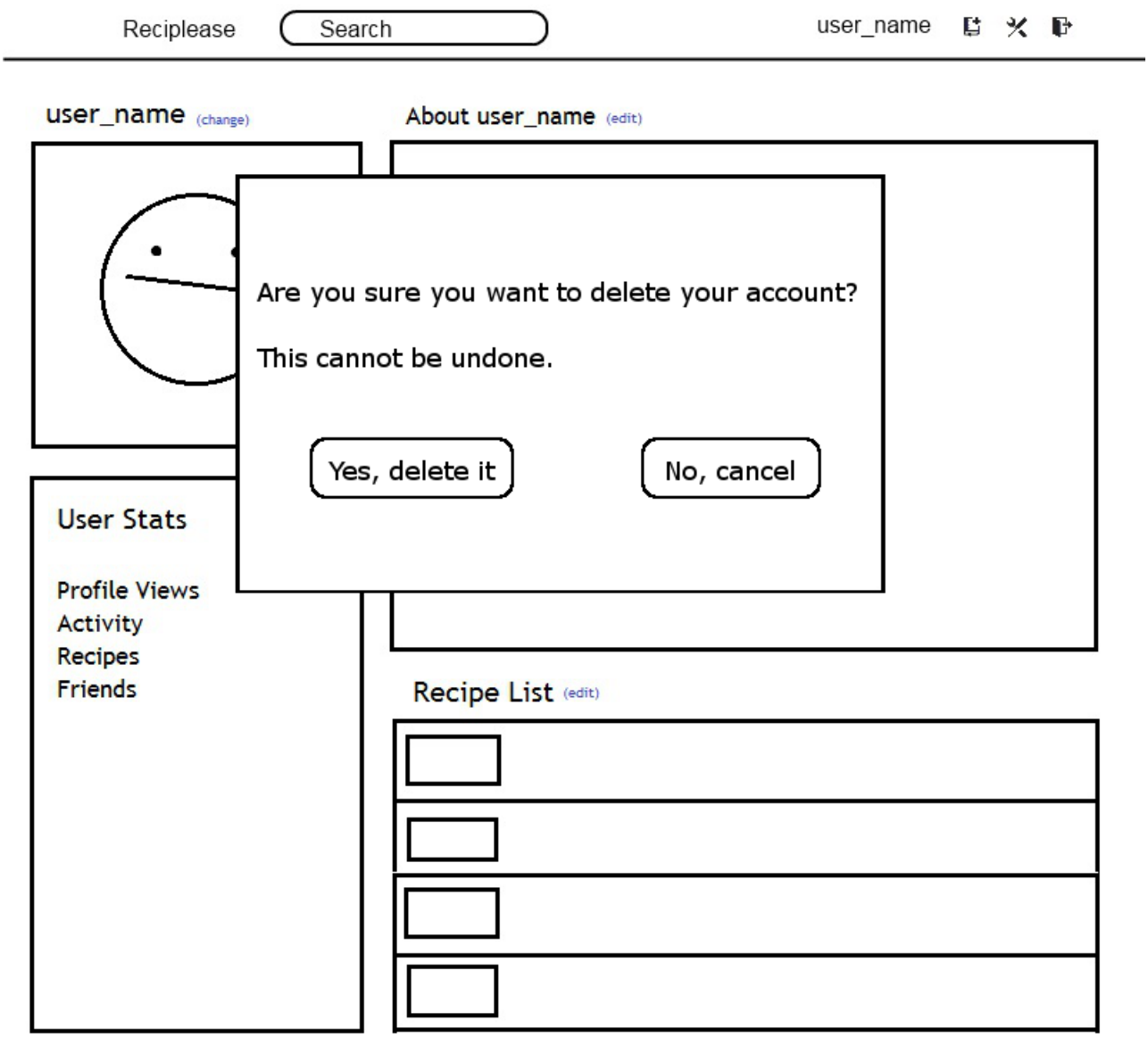


Delete

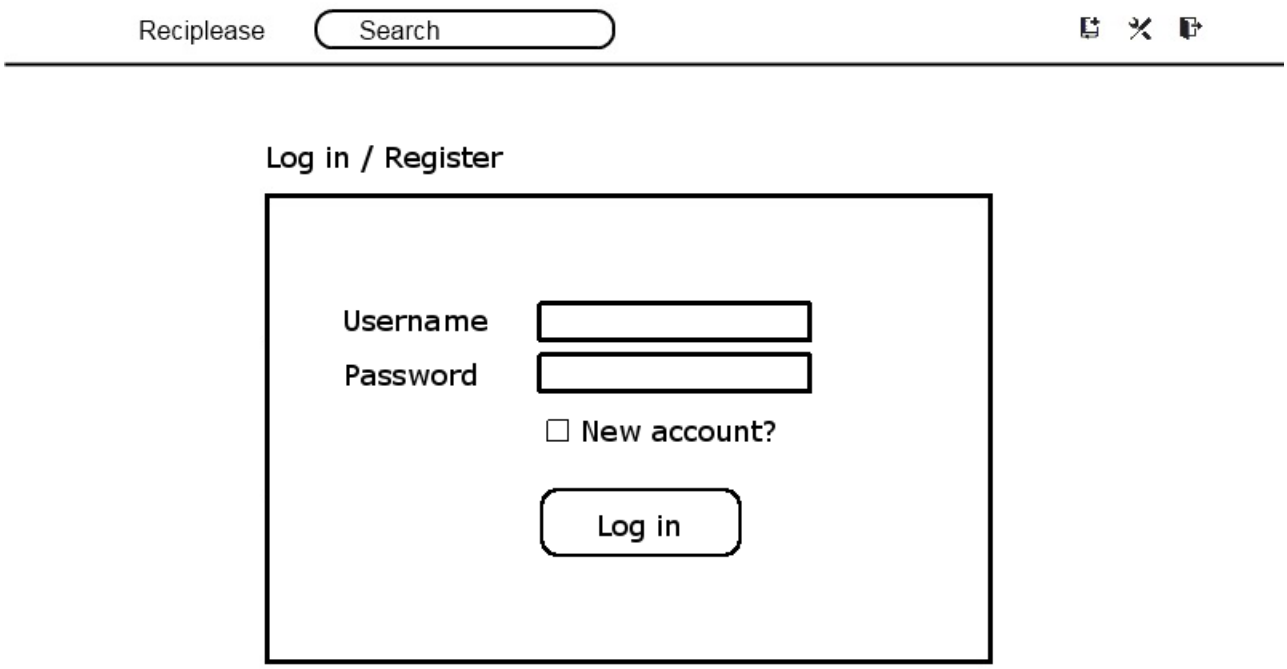
Delete Recipe



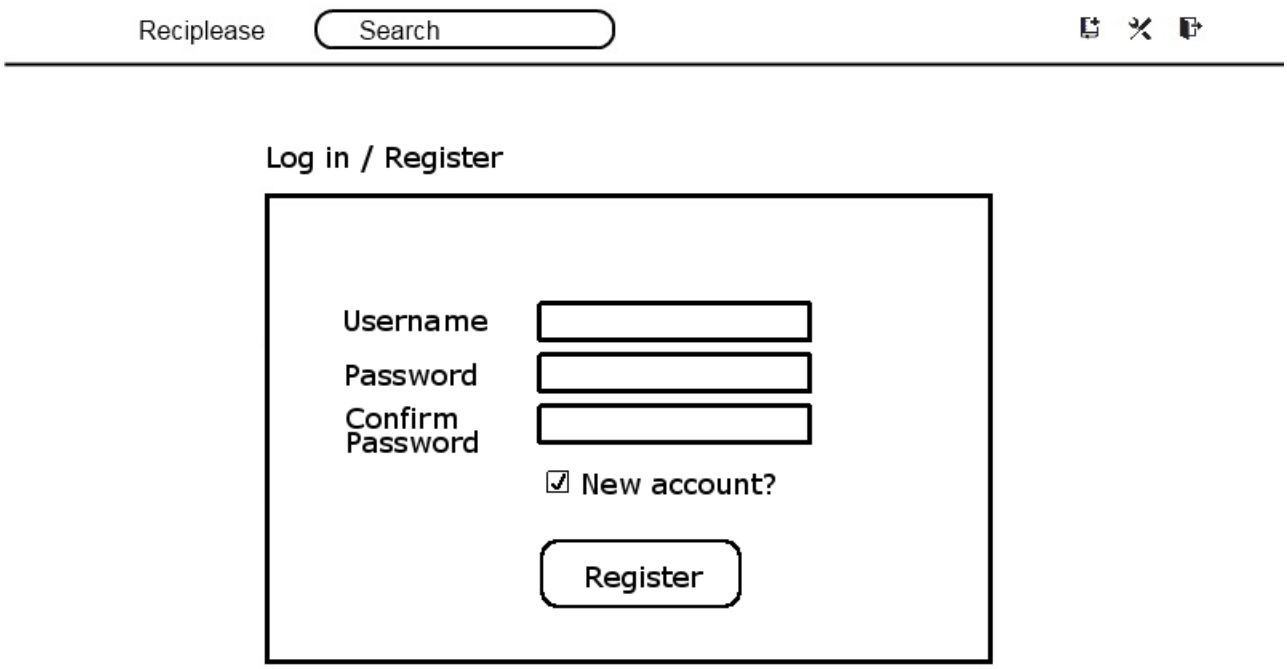
Delete User


Reciplease
Search
user_name
user_name (change)
About user_name (edit)
Are you sure you want to delete your account?
This cannot be undone.
Yes, delete it
No, cancel
User Stats
Profile Views
Activity
Recipes
Friends
Recipe List (edit)

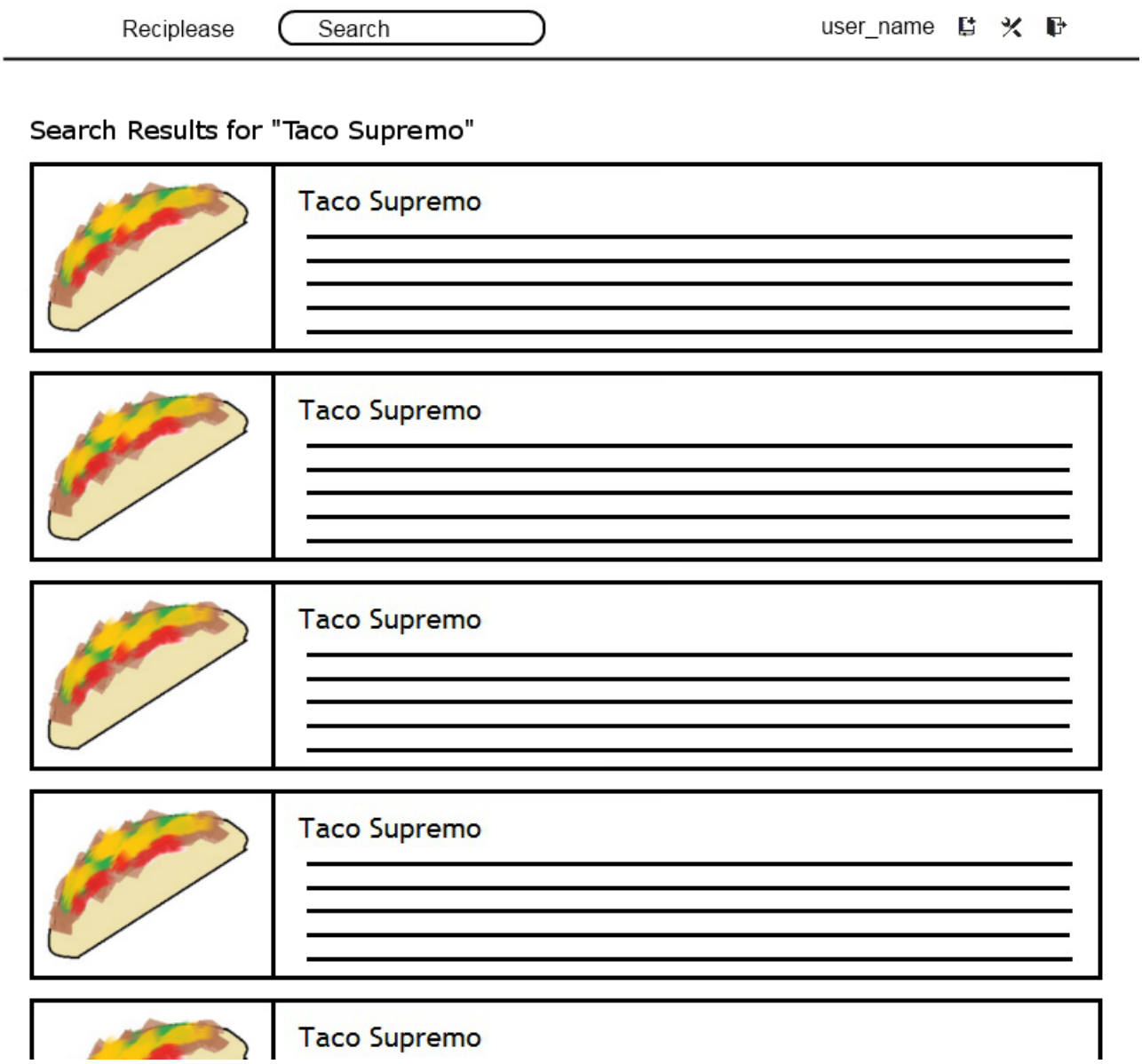
Log In


Reciplease
Search
Log in / Register
Username
Password
New account?
Log in

Register


Reciplease
Search
Log in / Register
Username
Password
Confirm Password
New account?
Register

Search


Reciplease
Search
user_name
Search Results for "Taco Supremo"
Taco Supremo
Taco Supremo
Taco Supremo
Taco Supremo
Taco Supremo

Glossary

API or application programming interface

A specification intended to be used as an interface by software components to communicate with each other.

Back End design

The design of databases and infrastructures that support websites and applications.

Cloud base hosting

Hosting an application in a cloud.

Cloud platform

A cloud platform is platform that delivers resources over a network.

Data flow diagrams

A graphical representation of the "flow" of data through an information system.

Fork

A fork happens when an individual takes some piece of information from another and starts independent work on that information creating a new unique piece of information.

Heroku

An cloud application platform

KLOC

1,000 lines of code

MTTR or Mean Time to Repair

The mean time to repair a system in the event that it should go down

Rails

An open-source web framework.

Ruby

Ruby is a dynamic, open source programming language.

Storyboards

A storyboard is a graphic in the form of illustrations or images displayed in sequence for the purpose of pre-visualizing the program.

Usability testing

A technique use to evaluate a product by testing it on users.

Use Cases

A list of steps, typically defining interactions between a user or external system and a system.

References

Email Questionnaire sent to Jimmy Theis on 9-9-2012

Meeting with Jimmy Theis and Bryan Watts on 9-12-2012

Managing Software Requirements: A Use Case Approach, Second Edition, by Dean Leffingwell and Don Widrig

http://www.ruby-lang.org/en/

http://www.heroku.com/

http://www.w3.org/

http://www.mustat.com/recipelabs.com/